REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-07-0343

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
26-07-2007	Final	March - July 2007

4. TITLE AND SUBTITLE
Knowledge Data Base for Amorphous Metals

5a. CONTRACT NUMBER
FA9550-07-C-0031

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hahn, Gail L.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Boeing Company, The
Engineering, Operations &
Technology - Phantom Works
JS McDonnell Blvd.
Saint Louis, MO 63166-0000

8. PERFORMING ORGANIZATION REPORT NUMBER

Requested

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

USAF, AFRL Duns143574726
AF Office of Scientific Res
875 N. Randolph St. Rm 3112
Arlington, VA 22203

Defense Adv Res Projects
Defense Science Office
3701 North Fairfax Drive
Arlington, VA 22203-1714

Dr Brett Conner/NA

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/PKR2
DARPA/DSO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A - Approved for public release; distribution unlimited PENDING WRITTEN APPROVAL

13. SUPPLEMENTARY NOTES
Major contributors to this program: David Giovannini, Object Computing Incorporated and Robert A. Bayles, Center for Corrosion Science and Engineering, Naval Research Laboratory

14. ABSTRACT
The program drew experts associated with thermal sprayed amorphous metals from the DARPA sponsored Naval Advanced Amorphous Coating (NAAC) program ( ), developers from the DARPA sponsored Accelerated Insertion of Materials initiative (N00421-01-3-0098), and computing specialists from Object Computing Incorporated to recommend a framework for a knowledge data base for amorphous metal coatings to reduce risk and accelerate transition of the technology among materials suppliers, laboratories/technical specialists, thermal spray operators, inspectors, and program managers. A workshop was held on 26 April 2007 for to explore needs, work shed the workflow/process, develop recommendations, and set priorities. Microsoft Access ™ was utilized to pilot the recommendations and further refine inspection sequences for the thermal spray process, and detail and demonstrate tracking (including shipping) and pedigree of the material, processes, coupons, plates, and coating installation on ship deck applications. The data base is for thermal spray coatings so it is applicable beyond NAAC.

15. SUBJECT TERMS
Amorphous Metals, Materials, Processing, Workflow, Database, Thermal Spray Coatings

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Joan Fuller
U	U	U	UU	13	19b. TELEPHONE NUMBER *(include area code)* 703-696-7236

This page is intentionally blank.

KNOWLEDGE DATA BASE FOR AMORPHOUS METALS

Project Objectives

The program drew experts associated with thermal sprayed amorphous metals from the DARPA sponsored Naval Advanced Amorphous Coating (NAAC) program, developers from the DARPA sponsored Accelerated Insertion of Materials initiative, and computing specialists from Object Computing Incorporated to recommend a framework for a knowledge data base for amorphous metal coatings to reduce risk and accelerate transition of the technology among materials suppliers, laboratories/technical specialists, thermal spray operators, inspectors, and program managers. A workshop was held on 26 April 2007 for to explore needs, work shed the workflow/process, develop recommendations, and set priorities. Microsoft Access ™ 2003 was utilized to pilot the recommendations and further refine inspection sequences for the thermal spray process, and detail and demonstrate tracking (including shipping) and pedigree of the material, processes, coupons, plates, and coating installation on ship deck applications. The data base is for thermal spray coatings so it is applicable beyond the amorphous metal coatings under study in the Naval Advanced Amorphous Coating program.

Work Carried Out

At the 26 April 2007 workshop, workflow and shipping were highlighted by customers as priorities. The main switchboard for the database, shown in Figure 1, was devised. ***Executive Summary*** provides a place for program managers to upload messages to the program team. ***Powder Supplier*** is for use by the powder supplier with interfaces for shipping information, chemical composition, characterization results by the supplier, and an interface for documenting experience with the use of the material, which is particularly important in development and transition periods for new technology. ***Powder Characterization*** has similar fields as Powder Supplier, however, it is used by laboratories independently characterizing the powder. ***Spray Provider*** is used by the thermal spray provider to document the application. ***Review of Documentation*** is used by the inspector in preparation of a spray application. ***Application Inspection*** is used by the inspector during witness of the application by the Spray Provider. The information from Spray Provider is also viewed on the inspector's interface in Application Inspection for ease of use. ***Coating Characterization, Non-Skid Test,*** and ***Materials Properties Test*** have not been developed. However, the intent is that they be developed in a very similar fashion to the pathfinder of ***Powder Characterization. In-Service Inspection*** provides the in-service inspector with a similar interface to that of Application Inspection; however, it is in-service rather than during coating application. To date, ***Reports*** only has a canned format for a shipping report as a pathfinder. However, the team has prepared a list of reports for prioritization and plans to continue this process as the database is socialized with the prospective user community.

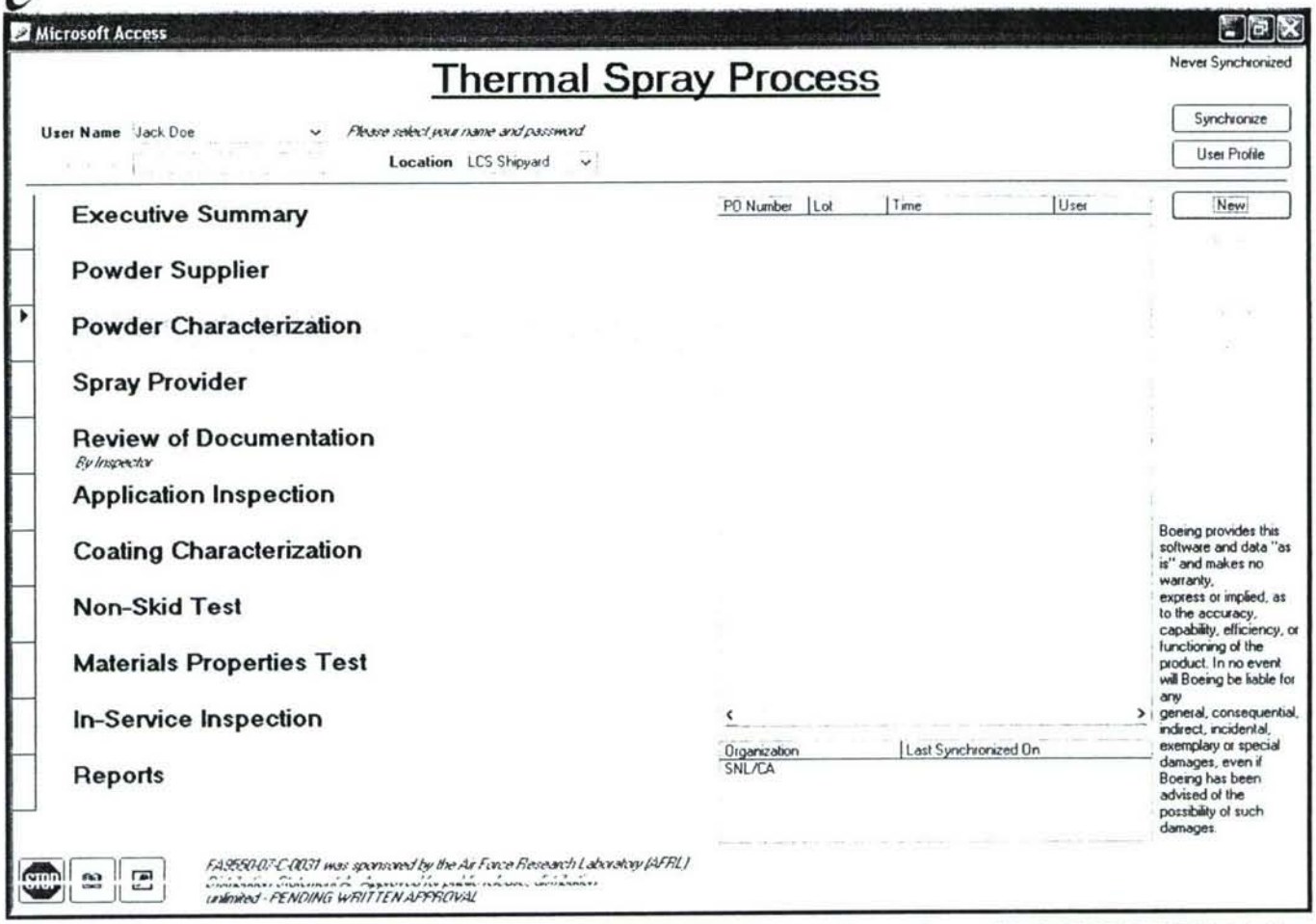


Figure 1. Main Switchboard of Thermal Spray Process Database (from file 20070726.3 NAAC.mdb)

Results Obtained

The database is available via three files: 20070726.3 NAAC.mdb, BigCheckBox.ocx, and KeyPadEdit.ocx.

NOTE: The 20070726.3 NAAC.mdb database requires installation of Microsoft Access™ 2003. After installing Microsoft Access, here are installation instructions for the database and its associated files: Copy the database file and the two accompanying OCX files, BigCheckBox.ocx and KeyPadEdit.ocx to a convenient place on your computer. The two OCX files must be in the same directory or folder as the database file. Launch the database, the .ocx files will self-install.

Appendix A, Thermal Spray Process Database Maintenance, covers the maintenance of the Microsoft Access ™ 2003 based database.

Appendix B, Server Synchronization, covers the issues associated with database synchronization to the Microsoft SQL™ Server. The application is designed to be thick-client and thin-server. The server is a Microsoft SQL™ Database. The database is the central repository of all data for the application. The client applications must regularly synchronize. Each client will have a complete local snapshot of the database.

Estimates of Technical Feasibility

Feasibility and refinement of workflow issues and database issues were carried out throughout the program via the Microsoft Access™ pilot database. The approach and software was reviewed by personnel from

government laboratories, technical people in government agencies and industry, material suppliers, spray providers, program managers, and information technology personnel.

In addition, a user interface was developed for a Tablet PC for ease of use by inspectors in the field. The Tablet PC (supplied by Naval Research Laboratory) with the interface was delivered to Naval Research Laboratory in July 2007. Two user interface controls were developed for Tablet PC use. The controls are independent OCXs developed in Visual Basic 6.0. The first is a resizable check box. The larger check box makes it easier to target the control with a stylus. The other control is a numeric entry edit field. It has a numeric keypad activated in a similar fashion as the drop-down list of a combo box as illustrated in Figure 2.

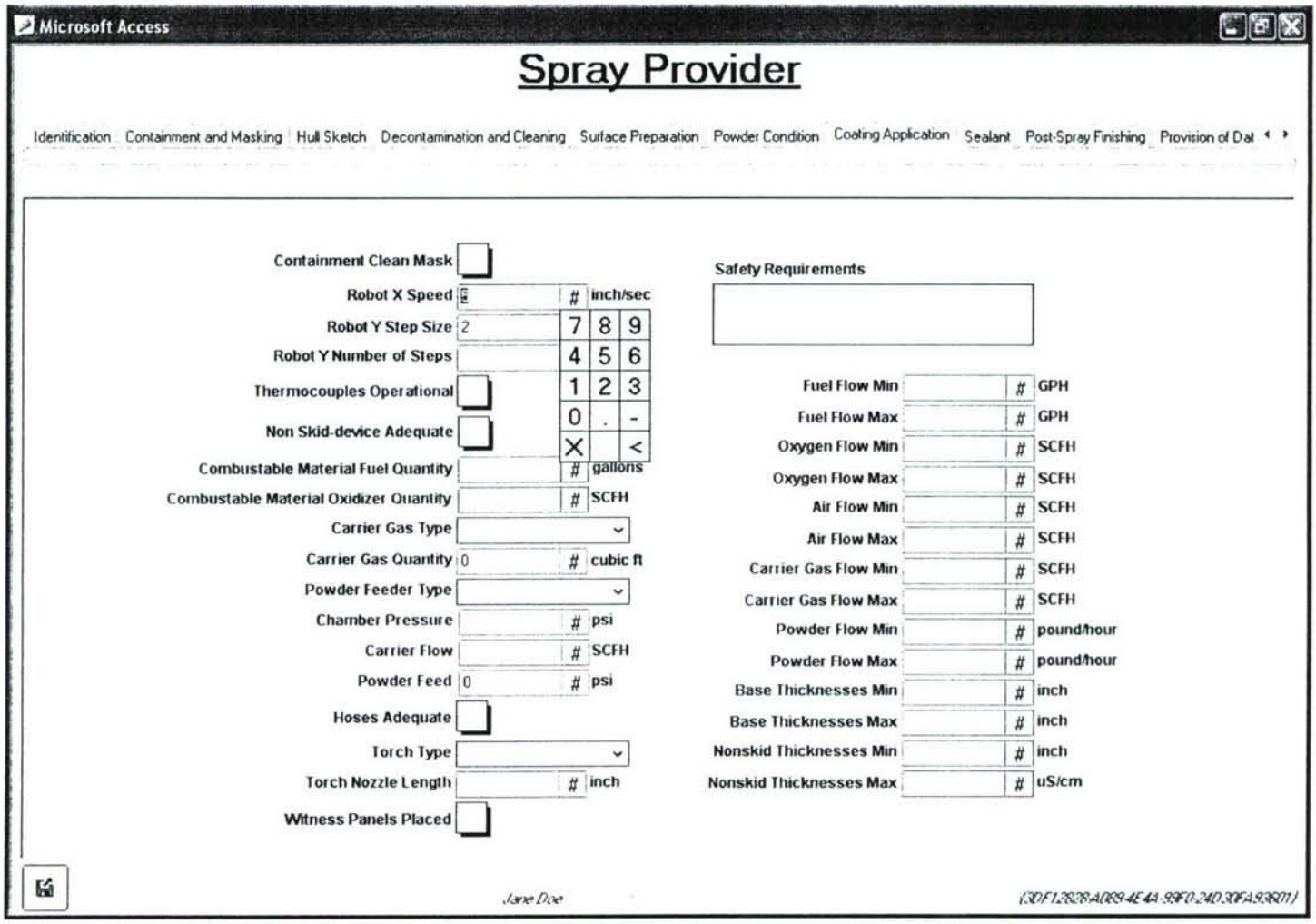


Figure 2. Example of Numeric Pad for User Ease

Object Computing Incorporated set up a Microsoft SQL™ Server for synchronization of data from powder suppliers, laboratories, spray providers, inspectors, and program management. Naval Research Laboratory is currently working through business and security issues to determine the best server approach for further development of the Microsoft Access™ based database developed in the this program as well as a web-based database for future consideration on other programs.

Microsoft (MS) Access provides a lightweight and strictly defined way of presenting data in the user interface. Anything outside of the preferred methods constitutes a work-around. At times MS Access user interface limitations could not be worked around. For instance, detail views of forms could not contain sub-forms, OCX controls, or independent controls. Microsoft is putting their development efforts into .NET and web- based solutions. MS Access has not seen a significant update in many years and likely never will again.

Instead, of Microsoft Access™, a web-based system is recommended for similar, new database developments.

- A web-based system may require no installation of software on the clients.
- If the clients need to operate without a connection to a server installation and data synchronization are required.
- Security is simplified by using secure web access.
- Modern Web-Apps provide easy cross-platform development
- Google Web Toolkit is recommended due to its high fidelity across platforms.

Among items that worked well in this database development program are the following:

- The data-driven workflow provided tools change the workflow via data, not programmatic, updates. Over 100 custom SQL statements that maintain the domain specific data are attached to the workflow entries in a generic fashion.
- Maintenance is simplified by defining behavior by database table entries, not writing code.

Among items that did not work well in this database development program are the following:

- This development effort suffered from not enough end-user and domain-expert testing. (The schedule called for testers from the Naval Advanced Amorphous Coating (NAAC) project in April. While there were participants at various points, they were not really in the capacity of domain-expert testers because of other priorities on the NAAC program itself.) Although the framework will change in future new programs, the methods of defining user interface behavior and data maintenance by data can carry to future projects.
- There is risk associated with data migration. The database is data-less at this time. It is generally difficult to train and motivate new users to input data. It is often even a greater hurdle to get legacy data input. Training, adequate funding, understanding of needs to device programs to assist data transfer must all be addressed.

Cumulative List of People Involved

People involved in this program:

Bob Bayles, Naval Research Laboratory
Jay Bourdreau, Government Consultant
Brett Conner, Air Force Office of Scientific Research
Joan Fuller, Air Force Office of Scientific Research
David Giovannini, Object Computing Incorporated
Gail Hahn, The Boeing Company
John Rivard, Strategic Analysis Incorporated, Consultant to Defense Advanced Projects Research Agency

Publications to Date

There have been no publications to date.

APPENDIX A
THERMAL SPRAY PROCESS DATABASE MAINTENANCE

The Thermal Spray Process database is a Microsoft Access-based application, MS Access™ 2003. When the database is opened, all menus, toolbars, and windows of MS Access are hidden. The form *Main Switchboard* is loaded and presents the lists of primary tasks to the user. To maintain the application, the user must login as a user with design privileges. An MS Access button will appear in the lower left next to the Stop button. When clicked, MS Access Graphical User Interface (GUI) elements reappear and the *Main Switchboard* closes.

The database GUI is presents a hierarchy of forms. When a task is selected from the main switchboard, a wizard or tabbed dialog appears presenting the user with a subform, or panel, for each step in the task. The user may exit and start new tasks, but the overall workflow is hierarchical.

The database's workflow is data-driven. There are four primary tables that determine application behavior: *Workflow*, *Workflow Run*, and *Run Generation*.

Workflow

The Workflow table defines the order of tasks and steps presented to the user. It is a hierarchy in which the root is the application. Modify the contents of this table to add new tasks or steps that will be visible in the GUI.

Hierarchy

The hierarchy is defined by the table's self-referential field, called parent_id. Every record (step) has a unique identifier (ID). Placing one step's ID into the parent id of another means the second step is now a component of the first. The steps are ordered by the number field. Currently, the steps that share a single parent **must** be contiguously numbered 1 to N.

The single record in which the ID and parent_id are equivalent is the root. The root represents the application itself. The immediate children of the root are used to populate the *Main Switchboard* form. These steps are the high-level tasks available in the application. When a user selects a task, the task's form is opened and a *workflow-run* is selected or created. A run contains the application data entered by the users.

Most tasks are then subdivided into panels for a wizard or tabbed dialog. Each panel has an entry in the Workflow table. A panel may further use the hierarchy in the workflow table for its specific needs.

The query _qry_workflow_outline may be used to present the entire hierarchy in an outline form.

Bindings

An individual step is implicitly tied to application data via the form_name field. This field tells the application what GUI form to load when this step is invoked. MS Access GUIs are bound to database tables via their record source and control source properties.
The form_name may not be unique. To present the same GUI in different contexts, simply add the same form_name to different steps. Application data may be prepared for the form by populating the SQL and run generation tables. Application data may be prepared in different ways for two steps that invoke the same form (and therefore bound to the same tables).

Workflow Run

A run contains all the data entered by the users for a single conceptual session. All the run's data may be spread across many application tables. A run has a Global Unique Identifier (GUID). This GUID, called Run Identifier

(RID), is used to link all application domain specific tables that save information for a specific run. Every run stores a time stamp, a user ID, its associated task, the last child step visited, and the RID that this run is based on (for edits, etc).

Nearly all application data is linked to the _workflow_run table. In an object-oriented point of view, anything that inherits from a workflow run shares a RID with an entry in the _workflow_run table.

Commands

The application supports 5 commands to invoke the workflow of a task in the switchboard: New, Continue, Edit, Review, and Redo.

- New generates a new run with empty or default field values.
- Continue uses a previous run and automatically returns to the last panel visited.
- Edit makes a complete copy and automatically returns to the last panel visited.
- Review simply shows the forms in a read-only manner.
- Redo copies only a small set of data and then creates new empty or default fields.

Run Generation

The Run Generation is the processes for preparing data for a run given the specified command (run generation mode). There is a single table of SQL commands. Each record has a user-definable ID, the SQL code, and a comment. The run generation table associates a set of SQL commands for a set given a specific mode.

A user selects a task, let's say "Coating Application," clicks "Edit," and all the SQL linked to this step and its child steps for the edit mode is immediately executed. The SQL, in this case, will copy the identification, powder characterization, and file records of the edited RID into a new RID.

SQL Table

The SQL table is simple a list of reusable SQL code to be executed. The SQL is parameterized. The old and new RID numbers are passed in (either of which may be NULL depending on the mode). The comments field is used to populate drop-down boxes in the maintenance GUIs. Each record has a user editable ID field. It is not an autonumber. The maintainer can sort and organize the SQL statements using an arbitrary numbering convention.

Some of the SQL can be difficult to maintain, like the copy SQL for a large table. The form _generateSQL may be used to produce generic SQL. Some tables, like chemical_analysis_values, cannot use the code generated by this form. But in most cases it saves a lot of time debugging SQL code.

Run Generation Table

The run generation table links a set of SQL records to a step for a given mode. The table, by itself, is not very maintainable. Each record is a set of four numbers. That is it. The _run_generation form is used to edit this table. This can be a dangerous form. Be careful. The form shows all the SQL for all modes of a single workflow step at a time. The last listed record is the new record, as with most MS Access data entry forms.

To create a new SQL/step association go to the last record.

- Select the run mode from the run mode popup.
- Change the invocation number if this is not the first SQL for this mode.
- Use the select popup to use an existing SQL command.
- Or enter in a new SQL ID number next to the popup to create a new SQL record.
- If this is new SQL, then edit the comment and sql boxes.

At any time you can change the SQL or comments in an existing run generation record. Any other step that uses that SQL will be altered as well.

Synchronization

The MS Access database synchronizes with a central server via ODBC. The synchronization is bi-directional. All workflow-run data is uploaded and downloaded. In other words, every MS Access database will contain all known data.

The specifications for what is synchronized are detailed in the _linked_tables table. Each record contains the client table name, the server table name, the ODBC link table name (representation of server table on client), and the SQL to execute for uploads and downloads.

When the user clicks synchronize from the main switchboard the _linked_tables table is queried, any missing ODBC links are created, downloads are executed, and then uploads.

Step-By-Step

The following set of instructions shows how to create a new task in the application.

1) Workflow

Start by defining the workflow of the task in the _workflow table. Create a new step whose parent ID is that of the root. Populate the fields with the title, info, number, and form name. You may need to edit the numbers of other tasks if this task is not the last item. The form name will likely be the "tabs" or "wizard," but need not be. Leave prerequisite as false and visible as true. The has_rid flag is used to tell the system if this task creates _workflow_runs.

_workflow : Table

ID	parent	number	title	info	form_name	prerequisite	visible	has_rid
1		0	Thermal Spray Process		main_switchboard	☐	☑	☑
2	1	1	Executive Summary		tabs	☐	☑	☐
31	1	2	Material Supplier		tabs	☐	☑	☑
32	1	3	Powder Characterization		tabs	☐	☑	☑
55	1	4	Spray Provider		tabs	☐	☑	☑
36	1	5	Application Inspection		wizard	☐	☑	☑
33	1	6	Coating Characterization		tabs	☐	☑	☑
34	1	7	Non-Skid Test		wizard	☐	☑	☑
35	1	8	Materials Properties Test		wizard	☐	☑	☑
65	1	9	In-Service Inspection		wizard	☐	☑	☑
66	1	10	Shipment Information		tabs	☐	☑	☐
44	2	1	Info		ppl_executive_info	☐	☑	☐

Record: 1 of 50

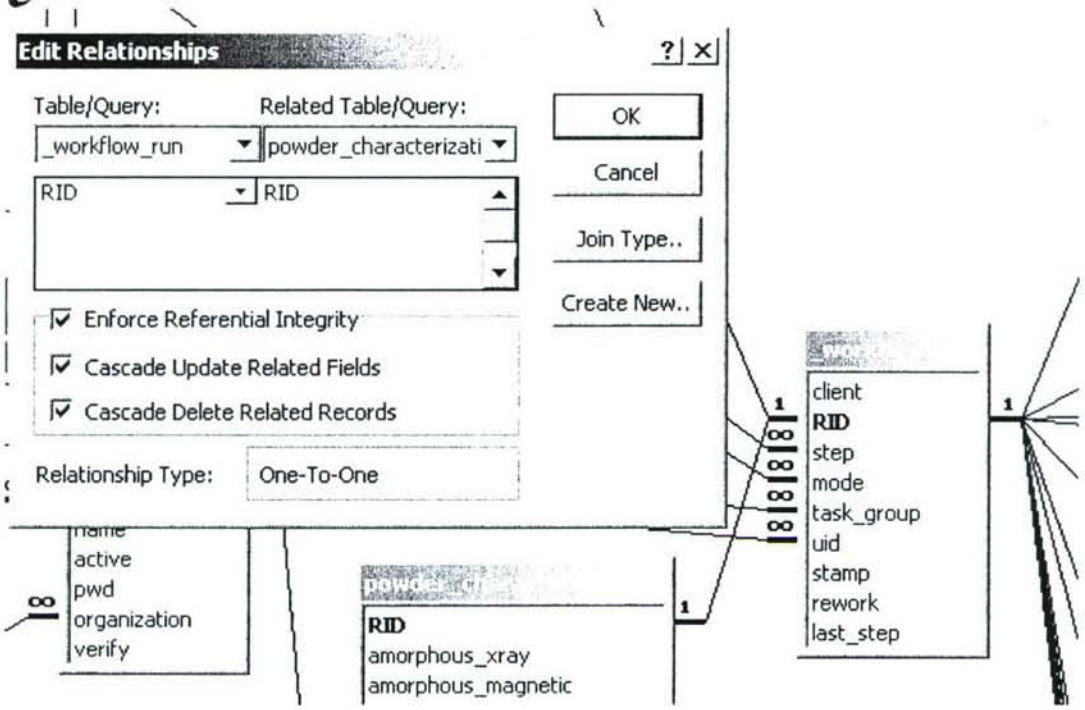
You will have to add an entry in _task_group_workflows for this new task to see it appear in the switchboard. You can disable specific commands for your task by editing the _enabled_commands table.

Now add the individual steps where the parent ID is that of the task. You don't have the form names yet, so leave those blank. Set prerequisite to true if the user cannot access steps after this until this step signals that it is ready.

Application Tables

Identify what existing tables can be reused and then create any new ones required. It is up to the maintainer's discretion how to normalize the tables.

All workflow run tables, though, must have a RID key. And this key must be linked to _workflow_run in the relationships with referential integrity and cascade updates/deletes.



When creating relationships, do not allow the workflow hierarchy determine the relationships. Powder Characterization has Chemical Analysis in its workflow but Chemical Analysis does not have a direct relationship with Powder Characterization. This is done so another workflow branch can use Chemical Analysis without bringing in a dependency on Powder Characterization.

Forms

If you have reused tables, likely you will reuse the same forms, but not necessarily so. Create the child step GUIs (panels). You may copy-paste existing panel forms to get started.

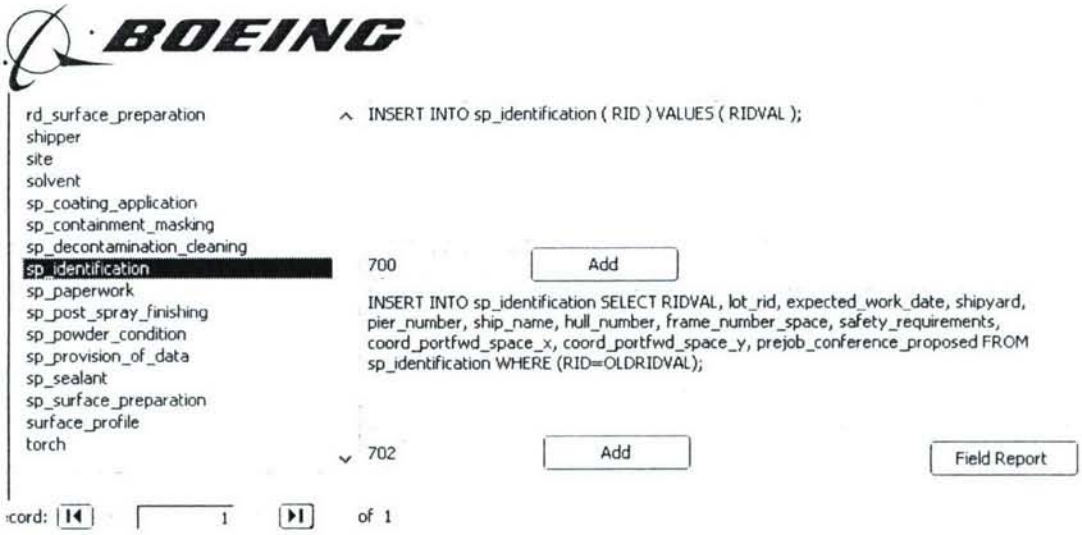
Each panel must implement the OnLinked method. Most panels have the only associated VBA code being an empty OnLinked method. Use prerequisite was set to true, the form's VBA must at some point invoke the PanelComplete method. See pnl_identification as an example.

Bind the forms and their fields to your application table(s). This is done using the Record Source and Control Source properties.

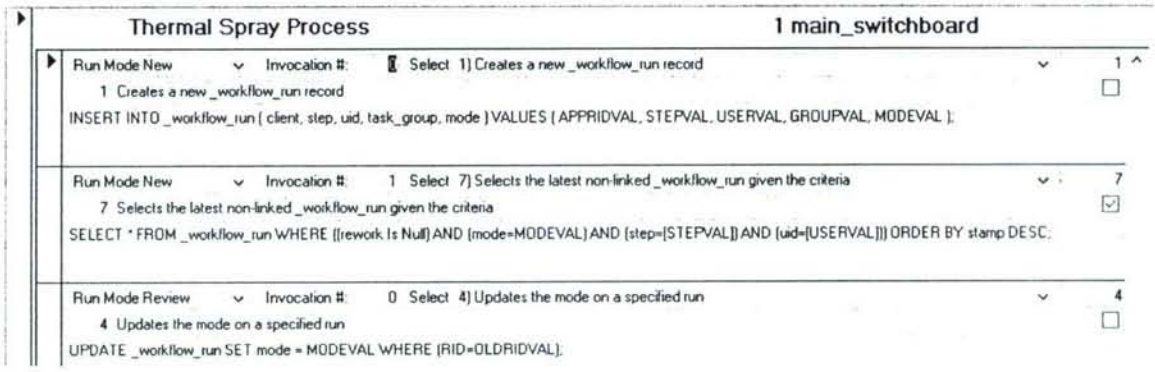
Now go back into the _workflow table and update the form_name felds appropriately.

SQL

We have now bound tables to forms, forms to steps, and steps to our application workflow. The last step is to train the application how to prepare the data. Use the _generateSQL and _run_generation forms to associate new or existing SQL to the new steps.
The _generateSQL form will quickly generate SQL to create a new record into a selected table and SQL to copy an existing record. Most tables can use the SQL as generated. You need to enter an ID for the SQL code. Be careful you enter the correct or appropriate IDS. Click the Add buttons to either insert or update sql records using the index provided.



Once the SQL is generated and recorded, they need to be linked to specific steps. Open the form _run_generation.



This form binds the SQL in the _SQL table to user actions in the applications work flow. Every step in the workflow has a page on this form. Use the navigation buttons on the bottom of the form to move to the workflow step needed to be modified.

Every detail line on the page was 3 primary fields, run mode, invocation # and selection. The run mode selects if the detail record binds the SQL to new, edit, review, etc. You can have more than one SQL commands for a step. The invocation number orders those steps. Finally the selection picks the SQL from the _SQL table. The _SQL's comment and code is editable here. If you change either, it will affect any other steps that use the same _SQL (as determined by the selection number).

APPENDIX B

SERVER SYNCHRONIZATION

The database application is a heavy client. Very little logic will reside on the server, besides the data repository. The server database will be Microsoft SQL™Server. The clients will connect to the server using SSL encryption/authentication and ODBC.

It is expected the users of the client database applications will regularly synchronize. A button on the application's main window will start the synchronization process. Information will be displayed stating the client's last synchronization date as well as others'.

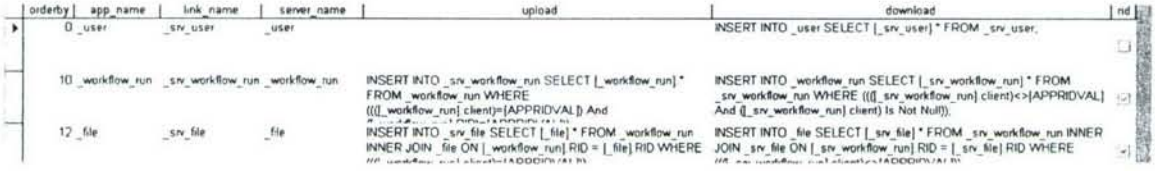
The synchronization both downloads and uploads all information. In other words, all clients and the server have the ability to hold all information collected from all clients.

Client Designations

Every client application, on its first run, will generate a unique identifier. All data uploaded to the server will be tagged with the client ID. All data is then traceable to a specific copy of the application. All data is tagged with a time-stamp and user identification as well.

Execution

A single table in the client application, *_linked_tables*, provides instructions for synchronization.



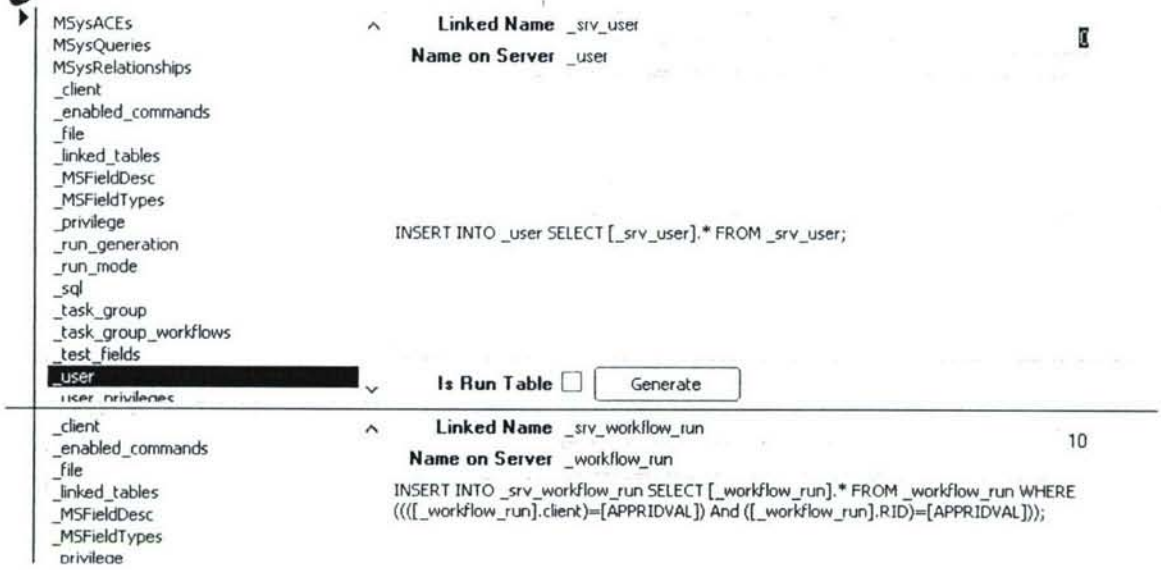
orderby	app_name	link_name	server_name	upload	download	nd
0	_user	_srv_user	_user		INSERT INTO _user SELECT [_srv_user].* FROM _srv_user;	
10	_workflow_run	_srv_workflow_run	_workflow_run	INSERT INTO _srv_workflow_run SELECT [_workflow_run].* FROM _workflow_run WHERE (([_workflow_run].client)=[APPRIDVAL]) And	INSERT INTO _workflow_run SELECT [_srv_workflow_run].* FROM _srv_workflow_run WHERE ((([_srv_workflow_run].client)<>[APPRIDVAL] And ([_srv_workflow_run].client) Is Not Null));	
12	_file	_srv_file	_file	INSERT INTO _srv_file SELECT [_file].* FROM _workflow_run INNER JOIN _file ON [_workflow_run].RID = [_file].RID WHERE	INSERT INTO _file SELECT [_srv_file].* FROM _srv_workflow_run INNER JOIN _srv_file ON [_srv_workflow_run].RID = [_srv_file].RID WHERE	

The *orderby* column simply designates order of execution. This helps ensure referential integrity. The three name columns designate the client table name, what the server's table name looks like to the client, and finally the real server table's name. The *upload* and *download* columns contain the SQL code necessary to perform the synchronization for the specific table. Downloads are performed first. The final column is checked for all tables associated with a workflow (stamped with client id, user, time, etc). *RID* stands for run-identification.

On user invoked synchronization, in order of the *orderby* column, the link is verified, the *download* SQL is executed, followed by the *upload*.

Maintenance

The SQL commands can be both unmanageably long and difficult to keep updated. A form, called *_generateLinkSQL*, can be used to auto generate the SQL commands.



Every entry in the table appears in this form. The left side list shows all tables in the client database. When a maintenance user clicks a new table, the linked and server names are populated with a best guess. The generate button will auto-generate the SQL code for upload and download.